Hybrid Modeling for Scenario-Based Evaluation of Failure Effects in Advanced Hardware-Software Designs

Jane T. Malin, NASA Johnson Space Center
Land Fleming, Hernandez Engineering, Inc.
David Throop, The Boeing Company

This paper describes an incremental scenario-based simulation approach to evaluation of intelligent software for control and management of hardware systems. A hybrid continuous/discrete event simulation of the hardware dynamically interacts with the intelligent software in operations scenarios. Embedded anomalous conditions and failures in simulated hardware can lead to emergent software behavior and identification of missing or faulty software or hardware requirements. An approach is described for extending simulation-based automated incremental failure modes and effects analysis, to support concurrent evaluation of intelligent software and the hardware controlled by the software.

Hybrid Modeling for Scenario-Based Evaluation of Failure Effects in Advanced Hardware-Software Designs

Jane T. Malin, Land Fleming and David Throop

Intelligent software for systems management has the promise of handling emergent situations and emergent requirements with flexible and robust operation, or emergent behavior. Such software will use its resources and get help from human operators, to make the best of unexpected problems and opportunities. It will be designed to autonomously assess a wide range of possible states and contexts and to select appropriate procedure variants that are consistent with goals. In anomalous fault management situations, it will use models of expected system behavior to detect and diagnose degradations and failures and identify appropriate recoveries, consistent with the situation and current goals. When presented with human instructions to perform tests and workarounds, it will use model-based prediction and what-if studies for self-validation of novel procedure variants before taking action.

A combination of test and evaluation approaches will be needed for validation of advanced hardware-software designs. The joint behavior of the operating hardware and software must be analyzed, to not only verify the software requirements but also discover missing requirements. Our experience with evaluation of complex intelligent software has shown that it can be difficult to envision effects of failures and recoveries in complex highly interconnected systems (Malin et al., 1998). Therefore, some software problems are actually the result of missing or faulty requirements.

We have been taking an incremental scenario-based simulation approach to evaluation of designs, software and requirements. Our hybrid models of hardware and operations have been used for evaluation of design of advanced gas processing systems and intelligent software for managing these systems autonomously (Malin et al., 2000). The CONFIG hybrid continuous/discrete event simulator provides capabilities to manipulate the structure of system models and to model failures of configuration, input, capacity, performance, control and operations. The hardware simulation dynamically interacts with the intelligent software in operations scenarios. Embedded anomalous conditions and failures in simulated hardware and control can lead to emergent software behavior. New requirements and design problems can also emerge in these scenarios.

We are developing an approach to model-based automated incremental failure modes and effects analysis (FMEA), using a function labeling approach similar to that developed by Price et al. (1995). Traditionally, FMEA is performed manually when design is nearly complete. It may drive design changes, but full FMEA is not performed on the revised design. The EPOCH project is intended to provide a capability to select failure models and assign function and malfunction labels to states of system components. Incremental FMEA uses a process of nested loops. Simulations are performed over a set of scripted operational scenarios, starting with a fault-free scenario. Times are logged when functions or malfunctions are reached or lost. Scripts are re-simulated for each failure mode case. Differences in the function logs (nominal vs. faulty) give the FMEA. The

FMEA is regenerated for each design alternative or change, and only failure effects that have changed are reported.

It should be possible to bootstrap the automated FMEA approach and get both types of analysis from the same models. Intelligent software or models of software operations can be run against multiple failure scenarios, to analyze effects of failure modes in operations and to evaluate the intelligent software. This will also support concurrent engineering of the system hardware and software, to take advantage of intelligent software capabilities. We have begun applying this approach to analysis of hardware-software designs for an In Situ Propellant Production System for exploration missions.

References

Malin, J. T., L. Fleming and T. R. Hatfield, "Interactive Simulation-Based Testing of Product Gas Transfer Integrated Monitoring and Control Software for the Lunar Mars Life Support Phase III Test." SAE Paper No. 981769. SAE 28th International Conference on Environmental Systems, Danvers MA, July 1998.

Malin, J. T., J. Nieten, D. Schreckenghost, M. MacMahon, J. Graham, C. Thronesbery, P. Bonasso, J. Kowing, and L. Fleming. "Multi-Agent Diagnosis and Control of an Air Revitalization System for Life Support in Space". IEEE Aerospace Conference. March 2000.

Price, C., D. Pugh, and J. Hunt, "Development of a Multiple Model Design Analysis System". Proc. 3rd Intl. Workshop on Advances in Functional Modeling of Complex Technical Systems, University of Maryland, June 1995.